

#4

PTO
1017 U.S.
09/022512
04/02/01

LAW OFFICES
SUGHRUE, MION, ZINN, MACPEAK & SEAS, PLLC
2100 PENNSYLVANIA AVENUE, N.W.
WASHINGTON, DC 20037-3213
TELEPHONE (202) 293-7060
FACSIMILE (202) 293-7860
www.sughrue.com

April 2, 2001

BOX PATENT APPLICATION
Commissioner for Patents
Washington, D.C. 20231

Re: Koutaro HACHIYA
METHOD AND APPARATUS FOR
MATRIX REORDERING AND ELECTRONIC
CIRCUIT SIMULATION
Our Ref. Q63926

Dear Sir:

Attached hereto is the application identified above including 30 sheets of the specification, including the claims and abstract, 9 sheets of informal drawings, executed Assignment and PTO 1595 form, and executed Declaration and Power of Attorney.

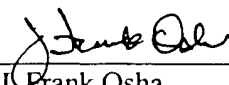
The Government filing fee is calculated as follows:

Total claims	20	-	20	=		x	\$18.00	=	\$0.00
Independent claims	9	-	3	=	6	x	\$80.00	=	\$480.00
Base Fee									\$710.00
Multiple Dependent Claim Fee									\$270.00
TOTAL FILING FEE									\$1460.00
Recordation of Assignment									\$40.00
TOTAL FEE									\$1500.00

Checks for the statutory filing fee of \$1460.00 and Assignment recordation fee of \$40.00 are attached. You are also directed and authorized to charge or credit any difference or overpayment to Deposit Account No. 19-4880. The Commissioner is hereby authorized to charge any fees under 37 C.F.R. §§ 1.16 and 1.17 and any petitions for extension of time under 37 C.F.R. § 1.136 which may be required during the entire pendency of the application to Deposit Account No. 19-4880. A duplicate copy of this transmittal letter is attached.

Priority is claimed from April 4, 2000 based on Japanese Application No. 2000-102163. The priority document is enclosed herewith.

Respectfully submitted,
SUGHRUE, MION, ZINN,
MACPEAK & SEAS, PLLC
Attorneys for Applicant

By: 
J. Frank Osha
Registration No. 24,625

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

K. Hachiya

4/2/01

① 63926

10f1

J1017 U.S. PTO
09/822512



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日

Date of Application:

2000年 4月 4日

出願番号

Application Number:

特願2000-102163

出願人

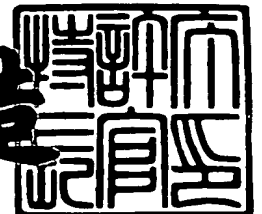
Applicant(s):

日本電気株式会社

2001年 2月16日

特許庁長官
Commissioner,
Patent Office

及川耕造



出証番号 出証特2001-3008136

【書類名】 特許願

【整理番号】 74510214

【提出日】 平成12年 4月 4日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 7/00

【発明の名称】 行列リオーダーリング方法及び装置並びに電子回路シミュレーション方法及び装置

【請求項の数】 8

【発明者】

【住所又は居所】 東京都港区芝五丁目7番1号 日本電気株式会社内

【氏名】 蜂屋 孝太郎

【特許出願人】

【識別番号】 000004237

【氏名又は名称】 日本電気株式会社

【代理人】

【識別番号】 100108578

【弁理士】

【氏名又は名称】 高橋 詔男

【代理人】

【識別番号】 100064908

【弁理士】

【氏名又は名称】 志賀 正武

【選任した代理人】

【識別番号】 100101465

【弁理士】

【氏名又は名称】 青山 正和

【選任した代理人】

【識別番号】 100108453

【弁理士】

【氏名又は名称】 村山 靖彦

【手数料の表示】

【予納台帳番号】 008707

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9709418

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 行列リオーダーリング方法及び装置並びに電子回路シミュレーション方法及び装置

【特許請求の範囲】

【請求項 1】 ガウス消去法を電子計算機によって並列実行して解を求める線形連立方程式の係数に対応する構造を有する係数行列の行列順序を決定する行列リオーダーリング方法であって、

係数行列中の非ゼロの要素の数と、ガウス消去法における累積的な処理時間に対応する値とに基づいて、第 1 の行及び列と、入れ替えるべき第 2 の行及び列を決定する第 1 の過程と、

第 1 の過程で決定された第 1 の行及び列と第 2 の行及び列とを入れ替える第 2 の過程と

を有することを特徴とする行列リオーダーリング方法。

【請求項 2】 ガウス消去法を電子計算機によって並列実行して解を求める線形連立方程式の係数に対応する構造を有する係数行列の行列順序を決定する行列リオーダーリング方法であって、

係数行列中の非ゼロの要素の数と、ガウス消去法におけるクリティカルパスの長さに基づいて、第 1 の行及び列と、入れ替えるべき第 2 の行及び列を決定する第 1 の過程と、

第 1 の過程で決定された第 1 の行及び列と第 2 の行及び列とを入れ替える第 2 の過程と

を有することを特徴とする行列リオーダーリング方法。

【請求項 3】 入れ替える行列の対称性に基づいて行列順序を入れ替える第 3 の過程と、

所定の条件で、前記第 1 及び第 2 の過程による行列順序の入れ替えと、第 3 の過程による行列順序の入れ替えとを選択する第 4 の過程と

をさらに有することを特徴とする請求項 1 又は 2 記載の行列リオーダーリング方法。

【請求項 4】 前記係数行列が非対称な構造を有する場合に、前記第 3 の過

程に対して、行列の構造を対称にする変換を行った後、入れ替えるべき行列を供給する第5の過程をさらに有することを特徴とする請求項3記載の行列リオーダーリング方法。

【請求項5】 ガウス消去法を電子計算機によって並列実行して解を求める線形連立方程式の係数に対応する構造を有する係数行列の行列順序を決定する行列リオーダーリング装置であって、

係数行列中の非ゼロの要素の数と、ガウス消去法における累積的な処理時間に対応する値とに基づいて、第1の行及び列と、入れ替えるべき第2の行及び列を決定する第1の手段と、

第1の手段によって決定された第1の行及び列と第2の行及び列とを入れ替える第2の手段と

を備えることを特徴とする行列リオーダーリング装置。

【請求項6】 ガウス消去法を電子計算機によって並列実行して解を求める線形連立方程式の係数に対応する構造を有する係数行列の行列順序を決定する行列リオーダーリング装置であって、

係数行列中の非ゼロの要素の数と、ガウス消去法におけるクリティカルパスの長さに基づいて、第1の行及び列と、入れ替えるべき第2の行及び列を決定する第1の手段と、

第1の手段によって決定された第1の行及び列と第2の行及び列とを入れ替える第2の手段と

を備えることを特徴とする行列リオーダーリング装置。

【請求項7】 電子回路に基づいて作成された線形連立方程式を解く過程を含んで電子回路のシミュレーションを行う電子回路シミュレーション方法において、

請求項1～4のいずれか1項に記載の行列リオーダーリング方法によって係数行列の行列順序を決定し、その行列順序を決定した線形連立方程式に対して、ガウス消去法を並列実行することで解を求める過程を有する

ことを特徴とする電子回路シミュレーション方法。

【請求項8】 電子回路に基づいて作成された線形連立方程式を解く手段を

有して電子回路のシミュレーションを行う電子回路シミュレーション装置において、

請求項 5 又は 6 項に記載の行列リオーダーリング装置によって係数行列の行列順序を決定し、その行列順序を決定した線形連立方程式に対して、ガウス消去法を並列実行することで解を求める手段を備える

ことを特徴とする電子回路シミュレーション装置。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は、線形連立方程式の係数行列の行列順序を決定する際に用いて好適な行列リオーダーリング方法及び装置、並びにそれを用いた電子回路シミュレーション方法及び装置に関する。

【 0 0 0 2 】

【従来の技術】

コンピュータを利用したシステムのシミュレーション方法には、システムを構成する複数の要素を線形連立方程式として記述し、これを繰り返し解くことで、システムの動作を解析するものがある。このようなシミュレーション方法において、線形連立方程式の解を求める方法の 1 つとしてよく利用されるものにガウス消去法がある。ガウス消去法では、 n 元 1 次連立方程式の各係数で構成される係数行列と定数項とを横に並べた形の n 行 $n + 1$ 列の行列を、上三角行列に変換する前進消去処理と、変換した行列の最下行から順次解を求める後退代入処理とによって、解が求められるようになっている。ここで、連立方程式の各係数で構成される係数行列内の各要素すなわち各元（特に非ゼロの元）を、所定の方法であらかじめ整理すること（行列リオーダーリングあるいは行列オーダリングすること）で、解を求める際に必要となる乗算や除算の回数を減らすことができることが知られている。

【 0 0 0 3 】

図 1 4 は、電子回路シミュレーションシステム等において用いられる従来の行列リオーダーリング部の処理の流れを示す流れ図である。一般に、従来の行列リオ

ーダリングにおいては、「最小次数法」と呼ばれている方法が採用されている（小国力編著、「行列計算ソフトウェア」、丸善株式会社等参照）。図 1 4 において、 N 行 \times N 列の係数行列のリオーダーリングを行う場合、変数 i に 1 を代入した後（S 1 0 1）、次に、与えられた係数行列のうちまだ順番が決定されていないピボットの中から最小次数のピボットを一つ選択し、その番号を変数 p に代入する（S 1 0 2）。ここで、第 k 番目のピボットとは、第 k 行の各元と第 k 列の各元をまとめて指すものであり（ k は 1 \sim N の整数）、次数とはピボット内の対角線上の要素を除き、対角線よりも列方向で下側または行方向で右側にある非ゼロ要素の個数を示し、そして、最小次数のピボットとは、 i 番目以降のピボットの次数で最小のものを有するピボットを指している。次に、選択した第 p 番目のピボットと第 i 番目のピボットとを交換することによって、第 i 番目のピボットを決定し（S 1 0 3）、第 i ピボットに対するガウス消去処理を行った時に発生する非ゼロ要素を係数行列に追加する（S 1 0 4）。ここで、第 p 番目のピボットと第 i 番目のピボットの交換とは、第 p 行の各元と第 i 行の各元をすべて交換するとともに、第 p 列の各元と第 i 列の各元をすべて交換すること、つまり行と列を同時に入替えることを意味する。これら一連の処理を、すべてのピボットの順序が決定するまで繰り返す（S 1 0 5, S 1 0 6）。「最小次数法」によれば、係数行列内で非ゼロ要素がより少ないピボットがより上側へと整理されるので、ガウス消去における乗算や除算の演算の総数を減少させることが可能となる。

【0 0 0 4】

【発明が解決しようとする課題】

ところで、上述したような従来の行列リオーダーリング方法によって整理した係数行列を用いて、複数のプロセッサを用いてガウス消去処理を並列実行する場合、単独のプロセッサで逐次実行した場合と比べて十分に高速化されないことがある。ガウス消去処理では、ある行の演算を行う場合に、他の行の演算結果を使用（参照）することになる。したがって、ガウス消去処理を並列実行する場合に、演算処理を行単位で複数のプロセッサに分けて並列処理するときには、演算の総数を減少させることと共に、各行の演算数と、各行で参照する他の行およびさらにその行で参照される他の行での演算数とを加えたものを減少させることを考慮

することが、高速化を図る上での重要な課題となる。しかしながら従来の行列リオーダーリング方法では、ガウス消去処理における総演算数の低減のみを行っていたため、ガウス消去処理を並列実行した場合に、必ずしも並列実行による高速化の効果を得ることはできなかった。

【 0 0 0 5 】

本発明は、上記の事情を考慮し、ガウス消去処理を複数のプロセッサを用いて並列実行する場合に、従来に比べてガウス消去処理の高速化を図ることができる係数行列を生成する行列リオーダーリング方法及び装置並びに電子回路シミュレーション方法及び装置を提供することを目的とする。

【 0 0 0 6 】

【課題を解決するための手段】

上記課題を解決するため、請求項 1 記載の発明は、ガウス消去法を電子計算機によって並列実行して解を求める線形連立方程式の係数に対応する構造を有する係数行列の行列順序を決定する行列リオーダーリング方法であって、係数行列中の非ゼロの要素の数と、ガウス消去法における累積的な処理時間に対応する値とに基づいて、第 1 の行及び列と、入れ替えるべき第 2 の行及び列を決定する第 1 の過程と、第 1 の過程で決定された第 1 の行及び列と第 2 の行及び列とを入れ替える第 2 の過程とを有することを特徴とする。請求項 2 記載の発明は、ガウス消去法を電子計算機によって並列実行して解を求める線形連立方程式の係数に対応する構造を有する係数行列の行列順序を決定する行列リオーダーリング方法であって、係数行列中の非ゼロの要素の数と、ガウス消去法におけるクリティカルパスの長さに基づいて、第 1 の行及び列と、入れ替えるべき第 2 の行及び列を決定する第 1 の過程と、第 1 の過程で決定された第 1 の行及び列と第 2 の行及び列とを入れ替える第 2 の過程とを有することを特徴とする。請求項 3 記載の発明は、入れ替える行列の対称性に基づいて行列順序を入れ替える第 3 の過程と、所定の条件で、前記第 1 及び第 2 の過程による行列順序の入れ替えと、第 3 の過程による行列順序の入れ替えとを選択する第 4 の過程とをさらに有することを特徴とする。請求項 4 記載の発明は、前記係数行列が非対称な構造を有する場合に、前記第 3 の過程に対して、行列の構造を対称にする変換を行った後、入れ替えるべき行列

を供給する第5の過程をさらに有することを特徴とする。

【0007】

請求項5記載の発明は、ガウス消去法を電子計算機によって並列実行して解を求める線形連立方程式の係数に対応する構造を有する係数行列の行列順序を決定する行列リオーダーリング装置であって、係数行列中の非ゼロの要素の数と、ガウス消去法における累積的な処理時間に対応する値とに基づいて、第1の行及び列と、入れ替えるべき第2の行及び列を決定する第1の手段と、第1の手段によって決定された第1の行及び列と第2の行及び列とを入れ替える第2の手段とを備えることを特徴とする。請求項6記載の発明は、ガウス消去法を電子計算機によって並列実行して解を求める線形連立方程式の係数に対応する構造を有する係数行列の行列順序を決定する行列リオーダーリング装置であって、係数行列中の非ゼロの要素の数と、ガウス消去法におけるクリティカルパスの長さに基づいて、第1の行及び列と、入れ替えるべき第2の行及び列を決定する第1の手段と、第1の手段によって決定された第1の行及び列と第2の行及び列とを入れ替える第2の手段とを備えることを特徴とする。請求項7記載の発明は、電子回路に基づいて作成された線形連立方程式を解く過程を含んで電子回路のシミュレーションを行う電子回路シミュレーション方法において、請求項1～4のいずれか1項に記載の行列リオーダーリング方法によって係数行列の行列順序を決定し、その行列順序を決定した線形連立方程式に対して、ガウス消去法を並列実行することで解を求める過程を有することを特徴とする。請求項8記載の発明は、電子回路に基づいて作成された線形連立方程式を解く手段を有して電子回路のシミュレーションを行う電子回路シミュレーション装置において、請求項5又は6項に記載の行列リオーダーリング装置によって係数行列の行列順序を決定し、その行列順序を決定した線形連立方程式に対して、ガウス消去法を並列実行することで解を求める手段を備えることを特徴とする。

【0008】

【発明の実施の形態】

以下、図面を参照して本発明による行列リオーダーリング方法およびそれを用いるシミュレーション方法の実施形態の一例について説明する。図1は、本発明を

電子回路のシミュレーションシステムに適用する場合の電子回路シミュレーション方法の全体の処理の流れを示す流れ図である。本実施の形態における電子回路のシミュレーションシステムは、線形連立方程式を解く過程を複数のプロセッサを並列稼働することによって行う電子計算機と、その電子計算機によって実行されるソフトウェアプログラムとによって実現されているものとする。また、そのプログラムはコンピュータ読みとり可能な記録媒体あるいはネットワークを介して頒布することが可能である。

【 0 0 0 9 】

まず、シミュレーション対象の電子回路を示す回路記述を、所定の入力装置から入力する（S 2 0 1）。次に、入力された回路記述に基づいて、回路を構成する各電子素子を示す線形方程式（1 次方程式）を、全素子に対応する連立方程式として構成する。そして、連立方程式の各変数の係数のうち、非ゼロである要素の位置を、行列 A に登録する（S 2 0 2）。ここで、行列 A は、電子回路に対応する連立方程式が N 元線形連立方方程式である場合に、N 行×N 列型の行列となる。ステップ S 2 0 2 では、この行列 A の各元（各要素）に、連立方程式の対応する係数が設定される。次に、行列リオーダーリング処理ステップ（S 2 0 3）では、行列 A 内の各要素の値を示す行列構造情報を用いて、行列 A の行の順序と列の順序を入替えることによって、ステップ S 2 0 6 の線形連立方程式の求解処理の処理時間が最小となるように行列順序を決定する。一般には、行列の第 i 行と第 i 列をまとめて第 i ピボットと呼ぶが、ここでは、このピボットを交換する処理を行う。つまり行と列を同時に入替えることによって行列順序を変更する。

【 0 0 1 0 】

次に、電子回路シミュレーションにおける最適な時間刻み幅 Δt を求め、次のタイムポイント t_{n+1} を、現在の時刻 t_n から Δt だけ進めた（ $t_{n+1} = t_n + \Delta t$ ）として決定する（S 2 0 4）。そして、ステップ S 2 0 3 で並べ替えが終了した行列 A の行列構造情報（行と列の順序）に基づき、その時刻における各素子の非線形特性を線形化してニュートン方程式（線形連立方程式）を作成する（S 2 0 5）。

【 0 0 1 1 】

次に、ステップ S 2 0 5 で作成したニュートン方程式をガウス消去法を適用して解く処理を行う (S 2 0 6)。本実施形態では、このニュートン方程式をガウス消去法を適用して解く処理を、複数のプロセッサを並列動作させることによって行う。次に、解が収束したかどうかを判定し (S 2 0 7)、収束しなかった場合には、ステップ S 2 0 5 へ戻って再度ニュートン方程式を作成し直してステップ S 2 0 6 の求解処理を行う。一方、ステップ S 2 0 7 で解が収束した場合には、時刻 t_n が最終時刻に達したかどうかを判定し (S 2 0 8)、達したとき ($t_n \geq$ 最終時刻の場合) には計算結果を出力して処理を終了し、まだ達していないときには現在の時刻 t_n を t_{n+1} に更新した後、ステップ S 2 0 4 へ戻って、上述したステップ S 2 0 4 ~ S 2 0 7 の処理を繰り返し実行する。

【 0 0 1 2 】

なお、上記のような電子回路のシミュレーション方法については、例えば、特開平 1 0 - 1 1 4 7 5 号公報「回路シミュレーション方法」、特開平 7 - 1 2 9 6 3 7 号公報「回路シミュレーション方法」等に記載されている。

【 0 0 1 3 】

図 1 に示すガウス消去処理 (S 2 0 6) は、連立方程式の変数を一つずつ消去していく過程とみなすことができる。変数 1 つを消去する処理を 1 つのタスクとして定義すると、各タスクの間には実行順序の制約がある。この制約は、例えば図 1 1 に示すような「タスクグラフ」と呼ばれる有向グラフによって示すことができる。図 1 1 に示す「タスクグラフ」では、「タスク i はタスク j の終了後に実行しなければならない」という制約を、「頂点 j から頂点 i に向かう辺がある」ことで表現している。図 1 1 において、頂点とは、番号 (数字) を丸で囲ったものであり、辺は矢印で示している。また、タスク i の「処理に要する時間」を頂点 i の「重み」として与えている。頂点の横の数字が頂点の「重み」を表している。

【 0 0 1 4 】

ガウス消去処理 (S 2 0 6) の並列実行時間は、このタスクグラフ内の任意の頂点から任意の頂点への経路のうち最も長い経路 (クリティカルパス) の長さで決まる。ここでいう「長さ」とは、「経路に含まれるすべての頂点の重みの総和

」である。したがって、本実施の形態が特徴とする行列リオーダーリング処理（S 2 0 3）では、このクリティカルパスをできるだけ短くするように行列のリオーダーリングを行う。

【0 0 1 5】

ここで、図1に示される行列リオーダーリング処理を説明するのに必要な「行列グラフ」、「ピボットの次数」、および「ピボットのクリティカルパス」について説明する。「行列グラフ」とは、行列Aの第*i*行、第*j*列の要素 $a(i, j)$ がゼロでない時、およびその時に限って、頂点*i*と頂点*j*の間に辺があるような無向グラフである。例えば、図6の行列の行列グラフは図7のようになる。グラフの頂点に繋がる辺の数は、一般に「次数」と呼ばれるが、行列グラフの頂点は行列のピボットに対応することから、頂点の次数を「ピボットの次数」とも呼ぶ。なお、図6において、×印と○印は非ゼロ要素を示し、対角線上に配置された○印で示す非ゼロ要素内の番号はリオーダーリング前のピボットの番号を示している。また、空白はゼロ要素を示し、記号「F」はガウス消去処理（S 2 0 6）において各ピボットの消去処理を行う際に追加される非ゼロ要素を示している。

【0 0 1 6】

$N \times N$ の正方向行列に対する行列リオーダーリング処理の途中で、1番目から*i*番目までのピボットのみの順序が決定され、*i* + 1番目から*N*番目までのピボットの順序は決定されていない時、ガウス消去処理では1番目から*i*番目の変数の消去処理は可能であり、したがってタスク1からタスク*i*の*i*個のタスクからなるタスクグラフを作成することができる。このタスクグラフ内のタスク*k*に至る任意の経路のうちで最長のものを「タスク*k*のクリティカルパス」と呼び、これを「第*k*番目のピボットのクリティカルパス」とも呼ぶことにする。また、タスク*k*のクリティカルパスに含まれるすべての頂点の重みの総和をタスク*k*あるいはピボット*k*の「クリティカルパス長」と呼ぶことにする。

【0 0 1 7】

図2は、図1に示す電子回路シミュレーションの流れ図における処理（S 2 0 3）である行列リオーダーリング処理の一例を示す流れ図である。まず、変数*i*に1を代入するとともに、変数*N*に行列Aの行列サイズを代入する（S 3 0 0）。

次に、全ての変数 k ($1 \leq k \leq N$) について、配列変数 $len(k)$ を $len(k) := 0$ と初期化する (S 3 0 2)。ここで、配列変数 $len(k)$ は、各時点で変化するピボット k のクリティカルパス長を記憶する変数である。次に、第 i 番目から第 N 番目のピボットの次数のうち、最小の次数 $mind eg$ を求める (S 3 0 4)。次に、第 i 番目から第 N 番目のピボットのうち、次数が $mind eg + \alpha$ 以下でかつクリティカルパス長が最小のピボット p を求める (S 3 0 6)。ここで、 α は 0 以上の所定の定数値であり、例えば α の推奨値は 1 から 3 の値となる。すなわち、ステップ S 3 0 6 では、次数が最小値 $mind eg$ から所定の範囲内にある近最小次数を有する 1 または複数のピボットのうちで、最もクリティカルパス長の小さい 1 つの第 p 番目のピボットを求めることになる。そして、第 p ピボットと第 i ピボットを入れ替える処理が行われる (S 3 0 8)。

【0 0 1 8】

次に、入れ替え後の第 i ピボットの消去処理で値が更新される全ての要素 e について、要素 e が属するピボット j のクリティカルパス長 $len(j)$ を、 $len(j) := \max \{ len(i) + w(i), len(j) \}$ となるように更新する (S 3 1 0)。すなわち、ステップ S 3 1 0 では、ステップ S 3 0 8 で第 p ピボットと第 i ピボットを入れ替えたことに影響される他のピボットのクリティカルパス長が最新の値に更新される。ここで、 $\max(x, y)$ は変数 x 、 y のうちで大きい方の値を示す関数であり、 $w(i)$ はピボット i の「重み」を与える関数である。そして、第 i ピボットの消去処理で発生する非ゼロ要素を係数行列 A に追加する (S 3 1 2)。次に、変数 i に 1 を加えることで更新し (S 3 1 4)、変数 i が変数 N を越えているかどうかを判定して (S 3 1 6)、 N を越えるまでステップ S 3 0 4 ~ S 3 1 2 の処理を繰り返し行う。以上の処理で、 $N \times N$ の行列 A すべてのピボットに対するリオーダーリングの処理が終了する。

【0 0 1 9】

なお、上記では、関数 $w(i)$ がピボット i の「重み」（処理に要する時間）を与える関数であると説明したが、関数 $w(i)$ は、図 1 におけるガウス消去法 (S 2 0 6) において、そのピボット i を消去する際に行われる例えば乗算および除算の数を加えたものに対応する値を得るものとして定義することができる。

例えば、ガウス消去処理を図5に示すようなプログラムで行う場合、 k 番目の変数を消去する処理では、重み $w(k)$ が、 $(deg(k) + 1) \cdot deg(k)$ 、すなわち第 k ピボットの次数 ($= deg(k)$) に 1 を加えたものに次数 ($deg(k)$) を掛けたものに対応する値 ($w(i) = (deg(i) + 1) \cdot deg(i)$) として求めることができる。なお、ガウス消去処理を行うためのプログラムの記述方法には、上記のもののほかにも種々のものがあり、異なるものを用いる場合には、上記の重みを求める関数 $w(i)$ も異なる記述による処理時間を反映するように適宜変更する。

【0020】

次に、図3を参照して、図2におけるステップS304とステップS306の処理についてより詳細に説明する。図3は、図2におけるステップS304とステップS306の処理の流れを示す流れ図であり、ステップS402～S412がステップS304内の処理に対応し、ステップS414～S426がステップS306内の処理に対応している。図3では、まず、最小次数を記憶する変数 $mindeg$ を初期化し (S402)、次に、変数 j に変数 i の値を設定する (S404)。そして、ステップS406～S412の処理を、変数 j が変数 N に達するまで繰り返し行う。ステップS406では、第 j 番目のピボットの次数 $deg(j)$ が変数 $mindeg$ より小さいか否かを判定し、小さい場合には変数 $mindeg$ を次数 $deg(j)$ で更新した後 (S408)、ステップS410で変数 j を $j := j + 1$ によって 1 だけ増加させる。そして、ステップS412で変数 j が変数 N より大きいかな否かを判定し、大きくない場合には再びステップS406以降の処理を実行する。以上の処理によって変数 $mindeg$ に最小の次数の値が格納される。

【0021】

次に、図2のステップS306に対応する処理では、まず、最短クリティカルパス長を記憶する変数 $minlen$ を初期化し (S414)、次に、変数 j に変数 i の値を設定する (S416)。そして、ステップS418～S426の処理を、変数 j が変数 N に達するまで繰り返し行う。ステップS418では、第 j 番目のピボットの次数 $deg(j)$ が (変数 $mindeg + \alpha$) より小さいかな否か

を判定し、小さい場合にはさらに第 j 番目のピボットのクリティカルパス長 $len(j)$ が変数 $minlen$ より小さいか否かを判定し (S 4 2 0)、小さいときには変数 $minlen$ をクリティカルパス長 $len(j)$ で更新するとともに、変数 p に変数 j を代入する (S 4 2 2)。そして、ステップ S 4 1 8 ~ S 4 2 2 のいずれかの処理が終了した後、ステップ S 4 2 4 で変数 j を $j := j + 1$ によって 1 だけ増加させる。そして、ステップ S 4 2 6 で変数 j が変数 N より大きいかな否かを判定し、大きくない場合には再びステップ 4 1 8 以降の処理を実行する。以上の処理によって、変数 p に第 i 番目から第 N 番目のピボットのうち、次数が $mindeg + \alpha$ 以下でかつクリティカルパス長が最小のピボット番号が格納されるとともに、変数 $minlen$ に最小のクリティカルパス長が格納される。

【 0 0 2 2 】

次に、本発明の行列リオーダーリング (図 1 のステップ S 2 0 3) 内の処理を図 2 に示す処理の流れに従って行う場合の具体的な動作について、図 8 ~ 図 1 1 を参照して説明する。なお、図 8 はシミュレーションしようとする電子回路の例、図 9 は図 8 の電子回路に対応する連立方程式の係数行列の例、図 1 0 は図 9 の行列のリオーダーリング後の行列の例、図 1 1 は図 1 0 に対応するタスクグラフをそれぞれ示している。

【 0 0 2 3 】

例えば図 8 に示すような 9 個の素子 $D 1 \sim D 9$ からなる電子回路を考える場合、電子回路内の各ノード $N 1 \sim N 7$ の電圧等の値を変数とする線形連立方程式の係数行列は、図 9 のように構成することができる。この場合、図 9 の係数行列は、第 i 番目のピボットが i 番目のノード番号に対応するように構成されている。図 9 では非ゼロ要素を \times 、ゼロ要素を空白で表現し、各要素の数値は省略してある。数字を囲んだ丸も非ゼロ要素を表しており、丸の中の番号は、リオーダーリング前のピボット番号を表す。第 i 変数のガウス消去処理 (タスク i) の重み ($w(i)$) は、(第 i 番目のピボットの次数 + 1) に (第 i 番目のピボットの次数) を掛けたものとする ($w(i) = (deg(i) + 1) \cdot deg(i)$)。以降では、 $\alpha = 0$ としたときの動作を説明する。

【 0 0 2 4 】

図 2 の流れ図において、まず $i = 1$ とし、次数が最小 ($\alpha = 0$ なので最小となる) のピボットで、かつそのピボットを第 i 番目のピボットと決定した時にそのピボットのクリティカルパス長が最小となるピボットを探す。図 9 において、第 1 ピボットと第 7 ピボットの次数は 1 で最小であり、かつこれらのクリティカルパス長はいずれも 2 で最小である。これらのうち番号の小さい第 1 ピボットを選択し、 $p = 1$ とする (S 3 0 6)。 $i = p$ であるので、ステップ S 3 0 8 でのピボット入れ替えは実際には行われぬ。また、第 1 ピボットについてガウス消去処理を行っても新たな非ゼロ要素は発生しない (S 3 1 2)。

【 0 0 2 5 】

続いて $i = 2$ として、まだ順序の決定されていない行列部分に対して上記処理を繰り返す。最小次数かつ最短クリティカルパスのピボットは第 7 ピボットであり $p = 7$ となる。第 2 ピボットと第 7 ピボットが交換される (S 3 0 8)。新たな非ゼロ要素は発生しない (S 3 1 2)。

【 0 0 2 6 】

$i = 3$ に対しては、最小次数・最短クリティカルパスのピボットとして $p = 3$ が選択される (次数は 2、クリティカルパス長は 6)。第 3 ピボットを消去すると新たな非ゼロ要素 $a(4, 7)$ 、 $a(7, 4)$ が追加される (S 3 1 2)。

【 0 0 2 7 】

$i = 4$ に対しては、最小次数・最短クリティカルパスのピボットとして $p = 5$ が選択される (次数は 2、クリティカルパス長は 6)。第 5 ピボットを消去すると新たな非ゼロ要素 $a(4, 6)$ 、 $a(6, 4)$ が追加される (S 3 1 2)。

【 0 0 2 8 】

$i = 5$ に対しては、最小次数・最短クリティカルパスのピボットとして $p = 4$ が選択される (次数は 2、クリティカルパス長は $6 + 6 = 12$)。第 4 ピボットを消去するときに新たな非ゼロ要素は発生しない (S 3 1 2)。

【 0 0 2 9 】

$i = 6$ に対しては、最小次数・最短クリティカルパスのピボットとして $p = 6$ が選択される (次数は 1、クリティカルパス長は $6 + 6 + 2 = 14$)。第 6 ピボ

ットを消去するとき新たな非ゼロ要素は発生しない（S 3 1 2）。

【 0 0 3 0 】

以上の処理によって、図 9 に示す係数行列が、図 1 0 に示すようにリオーダーリングされる。また、タスクグラフは図 1 1 に示すようになる。このとき、実行順序が最も遅いタスク 2 の次数は 0 であり、クリティカルパス長は 1 4 であり、すべての重みの合計は 2 4 である。

【 0 0 3 1 】

一方、従来の行列リオーダーリング（図 1 4）を図 9 の行列に適用すると、図 1 5 のような順序に並び替えられ、対応するタスクグラフは図 1 6 のようになる。この場合の実行順序が最も遅いタスク 2 のクリティカルパス長は 2 0 であり、すべての重みの合計は 2 4 である。

【 0 0 3 2 】

以上のように、図 1 の行列リオーダーリング（S 2 0 3）において、図 2 および図 3 を参照して説明したような、ピボットの近最小次数（近最小次数＝最小次数＋ α ）と、クリティカルパス長とを考慮した行列のリオーダーリングを行うことによって、本実施形態では、従来に比べて各タスクを複数のプロセッサによって並列実行した場合の処理時間をより短縮することが可能となる。

【 0 0 3 3 】

次に、図 1 の行列リオーダーリング処理（S 2 0 3）の他の実施形態、すなわち図 2 に示す構成の変形例について、図 4 を参照して説明する。図 4 は電子回路シミュレーションの流れ図（図 1）の処理ステップ（S 2 0 3）である行列リオーダーリング処理の流れ図である。おおまかには、図 2 ものと同様の近最小次数・最短クリティカルパス法（S 8）および従来の行列リオーダーリング方法の 1 つであるネステッド・ディセクション（nested dissection）法（S 1 0）の 2 つのリオーダーリング方法を選択可能な形で組み合わせたものとなっている。

【 0 0 3 4 】

「近最小次数・最短クリティカルパス法」による行列リオーダーリング部（S 8）は、変数 i を 1 に初期化した後（S 1）、与えられた係数行列 A のうちまだ順番が決定されていないピボットの中から近最小次数（次数が最小次数＋ α 以下、

α の推奨値は 1 から 3) かつ最短クリティカルパスを持つピボット (番号 p) を一つ選択し (S 2)、選択した第 p 番目のピボットと第 i 番目のピボットとを交換することによって第 i 番目のピボットを決定し (S 3)、第 i ピボットに対するガウス消去処理を行った時に発生する非ゼロ要素を係数行列 A に追加する (S 4)。これら一連の処理を、第 i ピボットの次数が定数 β を超えるか (S 5)、全てのピボットの順序が決定する (S 7) まで繰り返す。

【 0 0 3 5 】

上記の近最小次数・最短クリティカルパス法 (S 8) は、与えられた行列の平均次数が約 3 以下の時には、ネステッド・ディセクション法よりもクリティカルパス長を短くすることができるが、平均次数がそれより大きくなると、得られるクリティカルパス長はネステッド・ディセクション法によるものより長くなってしまう傾向がある。したがって、近最小次数・最短クリティカルパス法実行中に、選択されたピボットの次数が定数 β (β は自然数) より大きくなった場合には、ネステッド・ディセクション法へと切り替える。 β の値は 3 から 1 0 の範囲が適当である。

【 0 0 3 6 】

近最小次数・最短クリティカルパス法 (S 8) による行列リオーダーリング部の処理が終了してもまだ全てのピボットの順序が決定されていない場合には、係数行列 A 内のまだ消去されていない部分行列 R の行列グラフを作成し (S 9)、この行列グラフを用いてネステッド・ディセクション法により部分行列 R のみのリオーダーリングを行う (S 1 0)。「ネステッド・ディセクション法」は、行列グラフの 2 分割を再帰的に繰り返し、分割境界上の頂点に対応するピボットが最後にくるように順序付けを行う。すべてのピボットが順序付けられるまで、2 分割を再帰的に繰り返す処理が行われる。そして、部分行列 R に対するガウス消去処理で発生する非ゼロ要素を係数行列 A に追加する処理を実行して (S 1 1)、係数行列 A のリオーダーリングの処理が終了する。

【 0 0 3 7 】

次に、本発明の行列リオーダーリング (図 1 のステップ S 2 0 3) 内の処理を図 4 に示す処理の流れによって行う場合の具体的な動作について、図 6 ～図 9 およ

び図 1 2、図 1 3 を参照して説明する。この場合も、上記と同様に、図 8 に示すような 9 個の素子 $D_1 \sim D_9$ からなる電子回路と、リオーダリングの対象として図 9 に示すその線形連立方程式の係数行列を用いることとする。第 i 変数のガウス消去処理（タスク i ）の重み（ $w(i)$ ）は、上記と同様に、（第 i 番目のピボットの次数 + 1）に（第 i 番目のピボットの次数）を掛けたものとする。以降では、 $\alpha = 0$ 、 $\beta = 1$ としたときの動作を説明する。

【0038】

図 4 の近最小次数・最短クリティカルパス法（S 8）において、まず $i = 1$ とし、次数が最小（ $\alpha = 0$ なので最小となる）のピボットで、かつそのピボットを第 i 番目のピボットと決定した時にそのピボットのクリティカルパス長が最小となるピボットを探す。図 9 において、第 1 ピボットと第 7 ピボットの次数は 1 で最小であり、かつこれらのクリティカルパス長はいずれも 2 で最小である。これらのうち番号の小さい第 1 ピボットを選択し、 $p = 1$ とする（S 2）。 $i = p$ であるので、ステップ S 3 でのピボット入れ替えは実際には行われない。また、第 1 ピボットについてガウス消去処理を行っても新たな非ゼロ要素は発生しない（S 4）。

【0039】

続いて $i = 2$ として、まだ順序の決定されていない行列部分に対して上記処理を繰り返す。最小次数かつ最短クリティカルパスのピボットは第 7 ピボットであり $p = 7$ となる。第 2 ピボットと第 7 ピボットが交換される（S 3）。新たな非ゼロ要素は発生しない（S 4）。

【0040】

$i = 3$ に対しては、最小次数・最短クリティカルパスのピボットとして $p = 3$ が選択される（次数は 2、クリティカルパス長は 6）。第 3 ピボットを消去すると新たな非ゼロ要素 $a(4, 7)$ 、 $a(7, 4)$ が追加される。第 3 ピボットの次数は 2 であり $\beta = 1$ より大きいため、近最小次数・最短クリティカルパス法によるリオーダリングは終了される（S 5）。

【0041】

このとき、まだピボットの順番が決定されていない部分の行列 R は図 6 のよう

になっており、この行列グラフは図 7 のように生成される (S 9)。ネステッド・ディセクション法では、このグラフの 2 分割を行い、分割境界上のノードに対応するピボットが最後にくるように順序付けられる。すべてのピボットが順序付けられるまで 2 分割が再帰的に繰り返されるが、この例の場合は 1 度の 2 分割で十分であり、例えば頂点 4 と 6 が分割境界 P 1 上の頂点となり、5, 2, 4, 6 の順番で順序付けられる。最終的に得られる行列は図 1 2 となり、この行列に対するガウス消去処理のタスクグラフは図 1 3 のようになる。このタスクグラフの頂点の番号は図 1 2 のピボット番号に対応している。このタスクグラフのクリティカルパス長は 1 4 である。図 4 に示す行列リオーダリングの方法によっても、従来の行列リオーダリング (図 1 4) に比べてクリティカルパス長が減少していることがわかる。

【0 0 4 2】

なお、図 2 および図 4 を参照して説明した行列リオーダリング (図 1 のステップ S 2 0 3) の処理方法については、対称行列を対象とするものとして説明を行ったが、非対称行列に適用する場合には、次のような変更を行うようにすればよい。第 1 番目の変更点は、図 2 および図 4 の処理に共通するものであり、クリティカルパス長の最小のピボットを選択する際の必要条件である「次数 $\text{deg}(i)$ が所定の値 (α) 以下」という条件を、次数 $\text{deg}(i)$ の代わりに、 $\sqrt{\{NZLC(i) \cdot NZUR(i)\}}$ をパラメータとして用いて、「 $\sqrt{\{NZLC(i) \cdot NZUR(i)\}}$ が所定の値 (α) 以下」とする点である。ここで、 $NZLC(i)$ は、図 1 7 に示すような対角線上の要素を含まない下三角行列 L の第 i 列中の非ゼロ要素の数であり、 $NZUR(i)$ は、同じく対角線上の要素を含まない上三角行列 U の第 i 行中の非ゼロ要素の数である。

【0 0 4 3】

第 2 の変更点と第 3 の変更点は、図 4 に示す本発明による近最小次数・最小クリティカルパス法と、ネステッドディセクション法を組み合わせることで所定の条件に基づいて選択する場合の処理においてのみ適用される変更点である。図 2 に示す近最小次数・最小クリティカルパス法のみを用いる場合には、上記第 1 の変更点のみで非対称行列に対応可能である。第 2 の変更点は、図 4 に示す近最小次数・

最小クリティカルパス法（S 8）と、ネステッドディセクション法（S 1 0）との切り替えを行う条件を、次数 $\deg(i) > \beta$ （S 5）に代えて、第 1 の変更点と同様に、 $\sqrt{\{NZLC(i) \cdot NZUR(i)\}} > \beta$ と変更することである。そして、第 3 の変更点は、ネステッドディセクション法（S 1 0）を、非対称の行列 R の行列グラフにではなく、行列 $(R + RT)$ のように対称行列に変換し後の行列 $(R + RT)$ の行列グラフに適用することである。ここで、RT は、行列 R 行と列を入れ替えた転置行列である。これらの第 2 および第 3 の変更点を導入することで、図 4 に示す処理を非対称行列に適用することができる。

【0 0 4 4】

また、上記の変形例として、図 4 に示す処理において、非対称行列のリオーダーリングを行う場合には、あらかじめ、リオーダーリングの対象となる非対称係数行列 A とその転置行列 AT とから対称行列 $(A + AT)$ を作成して、これに対して、近最小次数・最小クリティカルパス法（S 8）と、ネステッドディセクション法（S 1 0）とによる並べ替えを行うようにすることもできる。

【0 0 4 5】

なお、本発明の行列リオーダーリング方法の適用は、電子回路のシミュレーション方法に限定されるものではなく、1 次連立方程式をガウス消去法によって並列実行して解く過程を含むデータ処理方法であれば、それに適用することができる。

【0 0 4 6】

【発明の効果】

以上説明したように、本発明によれば、従来に比べて、ガウス消去処理のタスクグラフのクリティカルパス長を低減しているため、ガウス消去処理を並列実行した際の実行時間を短縮することができる。

【図面の簡単な説明】

【図 1】本発明による電子回路シミュレーション方法の一実施形態を示す流れ図。

【図 2】図 1 に示す電子回路シミュレーション方法における行列リオーダーリング処理（S 2 0 3）の流れの一実施形態を示す流れ図。

【図 3】図 2 のステップ S 3 0 4 , S 3 0 6 の処理の流れを示す流れ図。

【図 4】図 1 に示す電子回路シミュレーション方法における行列リオーダーリング処理 (S 2 0 3) の流れの他の実施形態を示す流れ図。

【図 5】図 1 のガウス消去法 (S 2 0 6) の計算方法を記述したプログラムの一例を示す説明図。

【図 6】まだ順序づけされていない部分行列 R を示す図。

【図 7】図 6 の行列に対するガウス消去処理のタスクグラフ。

【図 8】入力される回路例を示す回路図。

【図 9】図 8 の回路図に対応するリオーダーリング前の行列を示す図。

【図 1 0】図 9 の行列を、図 2 に示す行列リオーダーリング処理 (近最小次数・最小 (最短) クリティカルパス法) によってリオーダーリングした後の行列を示す図。

【図 1 1】図 1 0 の行列リオーダーリング後の行列についてのタスクグラフ。

【図 1 2】図 9 の行列を、図 4 に示す行列リオーダーリング処理によってリオーダーリングした後の行列を示す図。

【図 1 3】図 1 2 の行列リオーダーリング後の行列についてのタスクグラフ。

【図 1 4】従来の行列リオーダーリング処理 (最小次数法) によるリオーダーリング処理の流れを示す流れ図。

【図 1 5】図 9 の行列を、図 1 4 に示す従来の行列リオーダーリング処理によってリオーダーリングした後の行列を示す図。

【図 1 6】図 1 5 の行列リオーダーリング後の行列についてのタスクグラフ。

【図 1 7】上下三角行列を説明するための図。

【符号の説明】

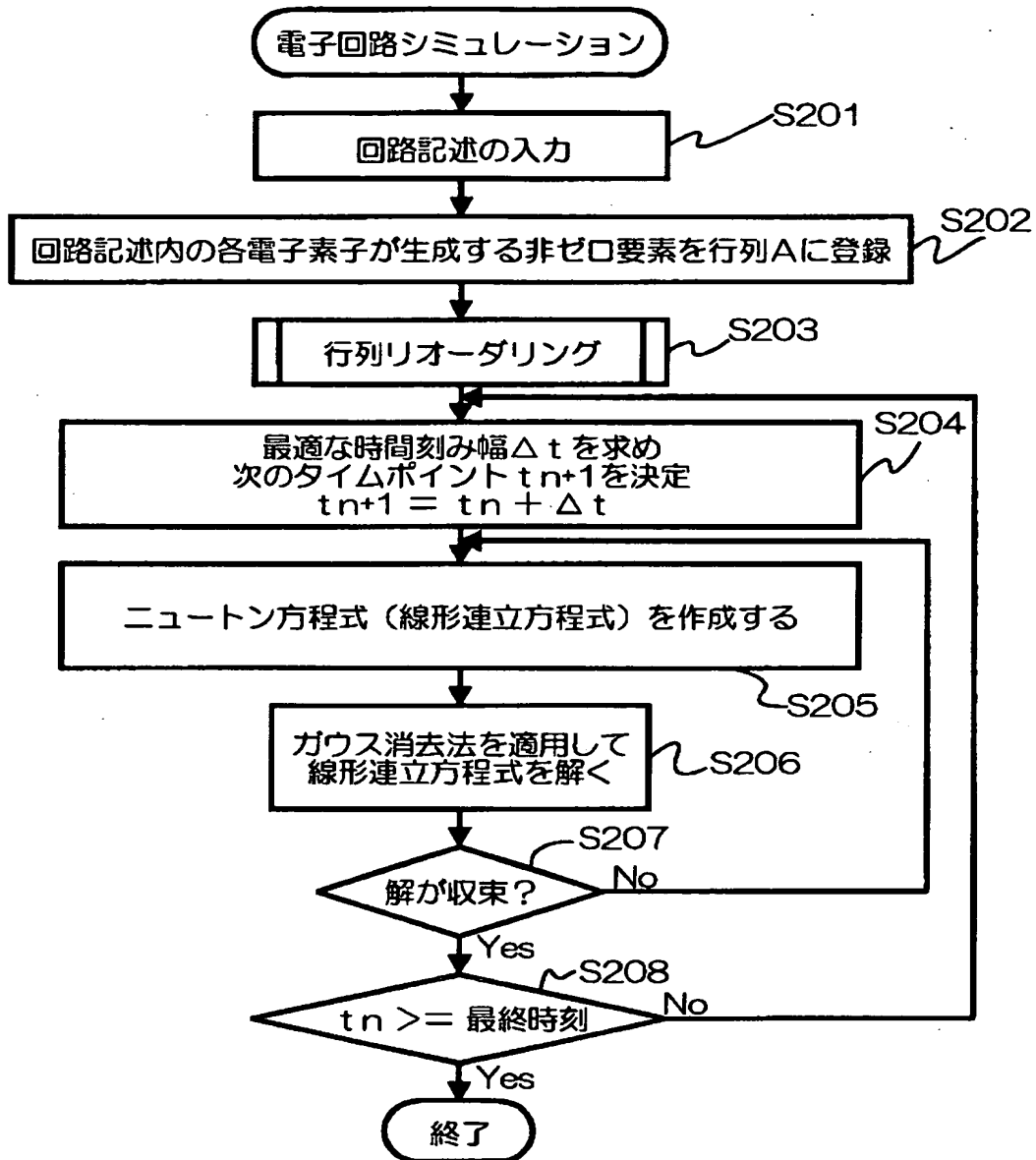
S 2 0 3 行列リオーダーリング処理

S 2 0 6 ガウス消去法を適用する線形連立方程式の求解処理

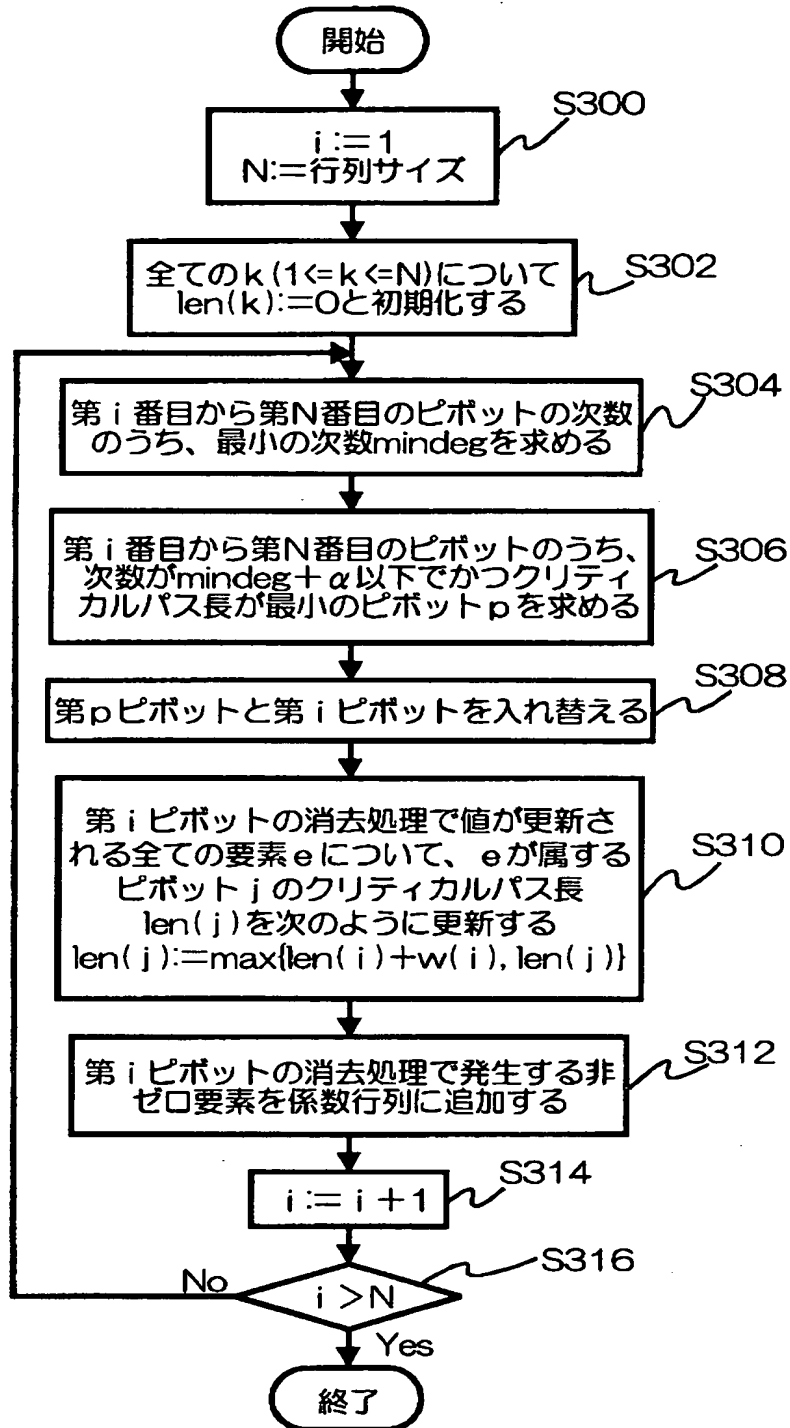
【書類名】

図面

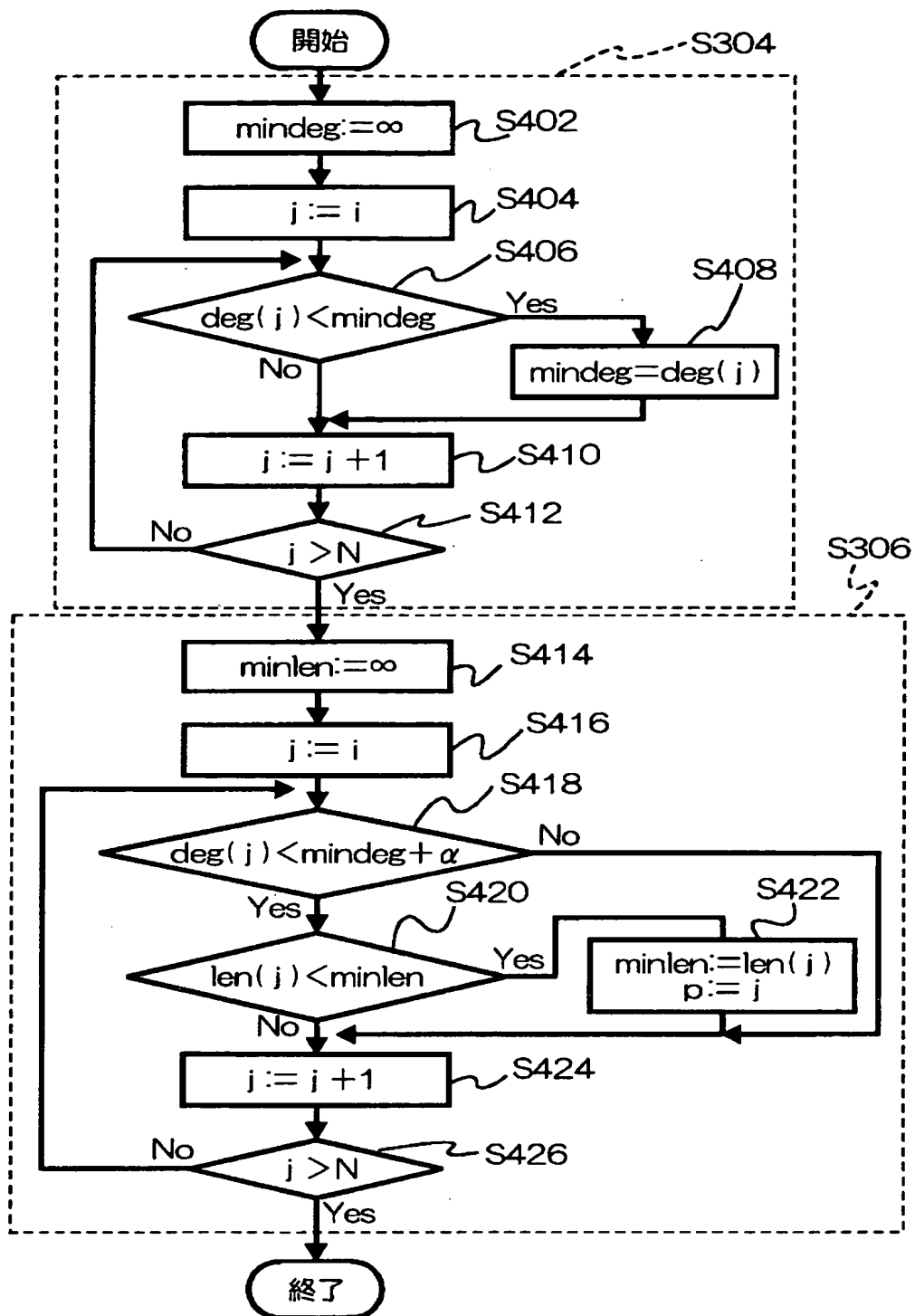
【図 1】



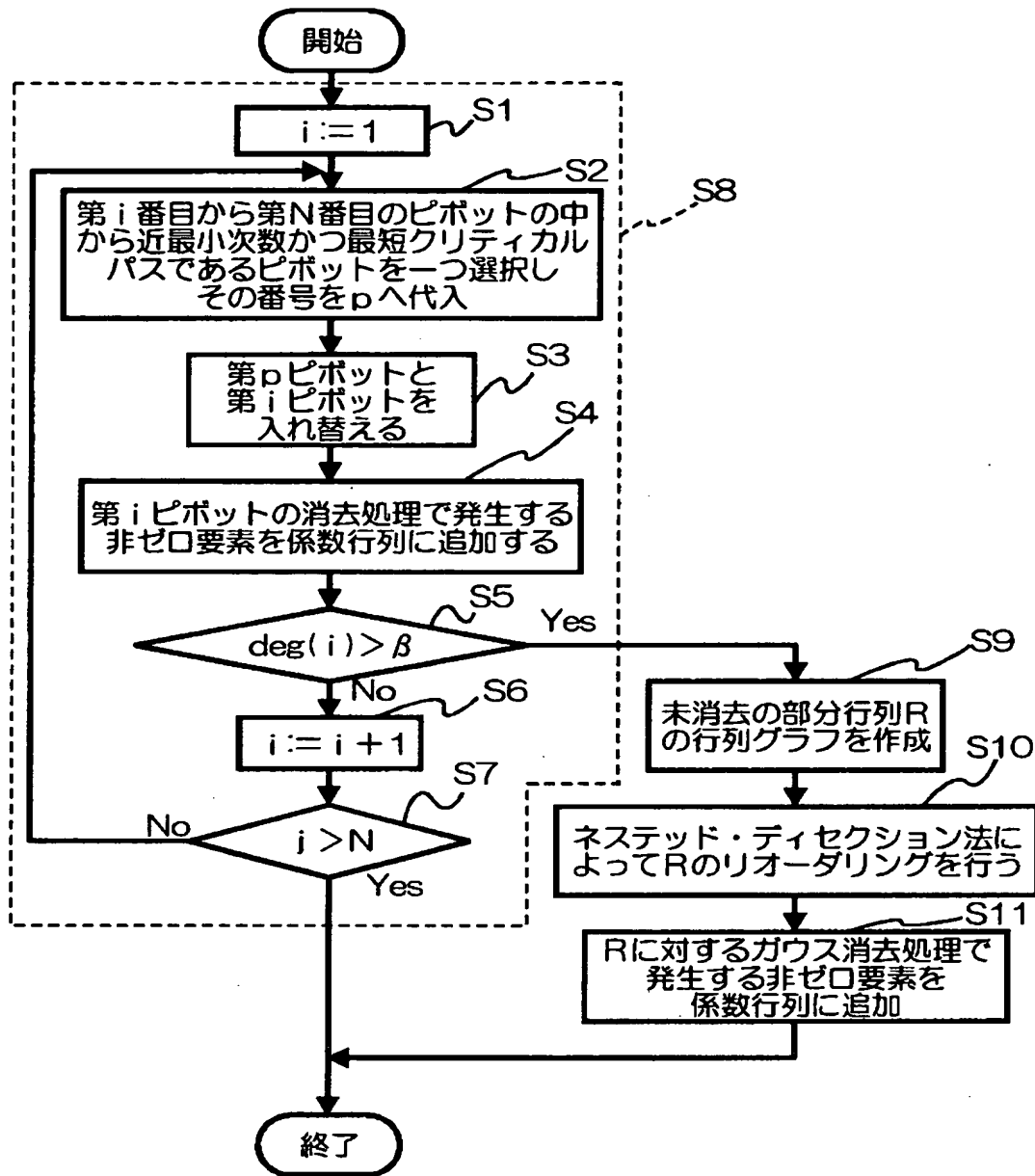
【図 2】



【図 3】



【図4】



【図 5】

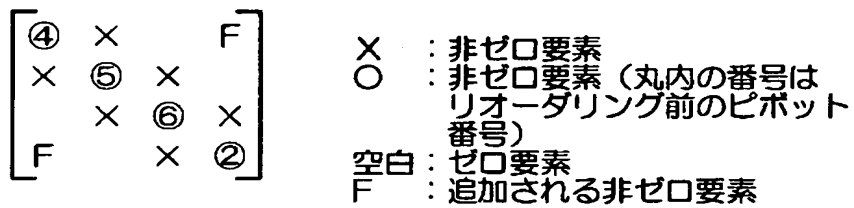
```

for k=1:n-1
    for i=k+1:n
        if a(i,k) <> 0 then
            a(i,k) = a(i,k) / a(k,k)
            for j=k+1:n
                if a(k,j) <> 0 then
                    a(i,j) = a(i,j) - a(i,k) * a(k,j)
                end
            end
        end
    end
end
end

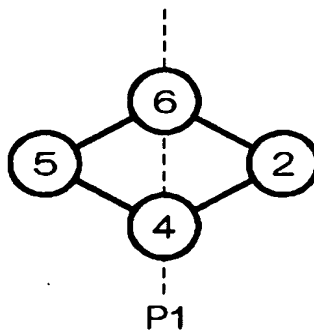
```

k 番目の変数を消去
 除算
 乗算

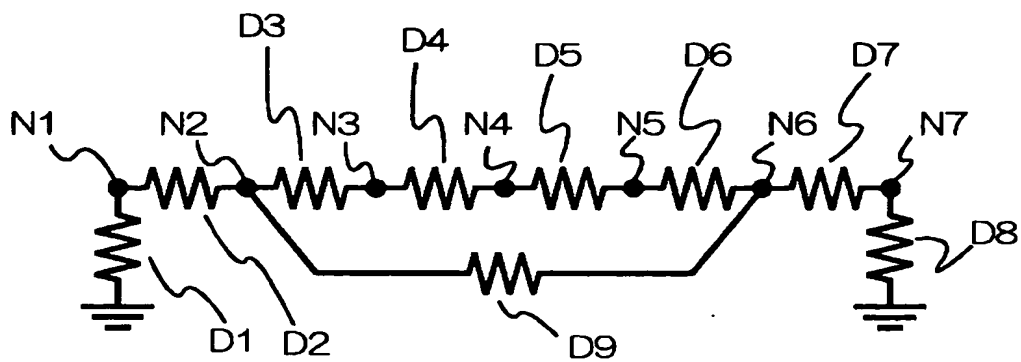
【図 6】



【図 7】



【図 8】

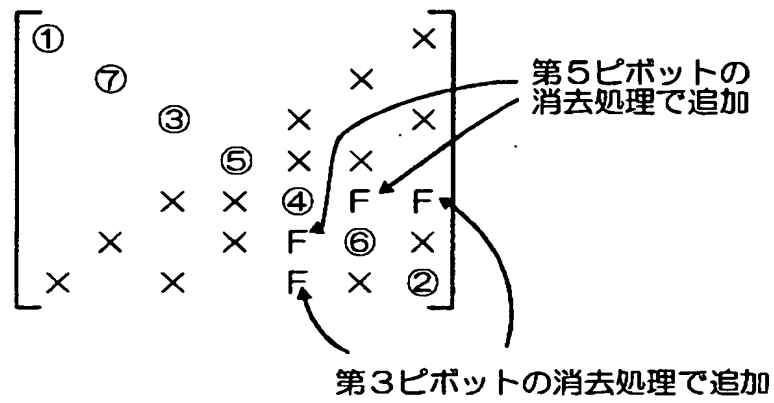


【図 9】

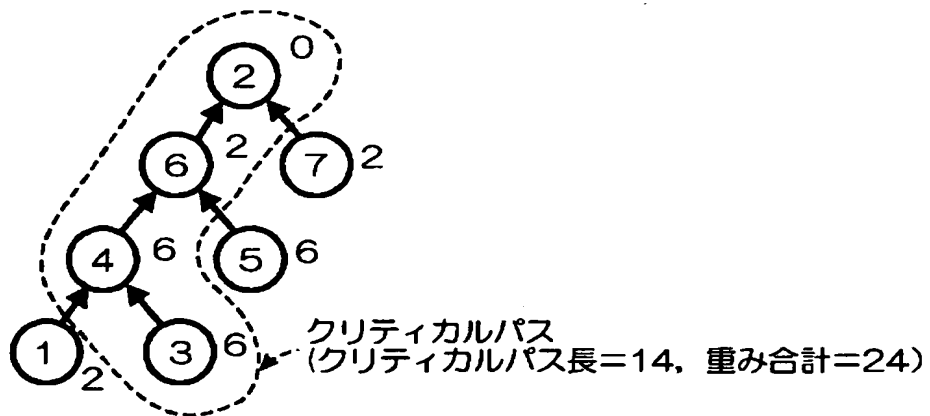
$$\begin{bmatrix} \textcircled{1} & \times & & & & & \\ \times & \textcircled{2} & \times & & & \times & \\ & \times & \textcircled{3} & \times & & & \\ & & \times & \textcircled{4} & \times & & \\ & & & \times & \textcircled{5} & \times & \\ & \times & & & \times & \textcircled{6} & \times \\ & & & & & \times & \textcircled{7} \end{bmatrix}$$

X : 非ゼロ要素
 O : 非ゼロ要素 (丸内の番号は
 リオーダーリング前のピボット
 番号)
 空白 : ゼロ要素

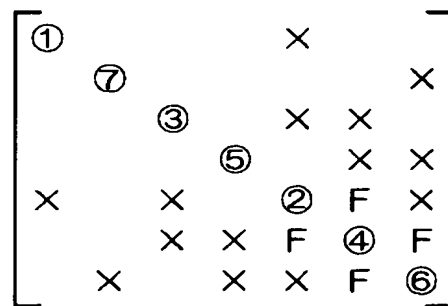
【図10】



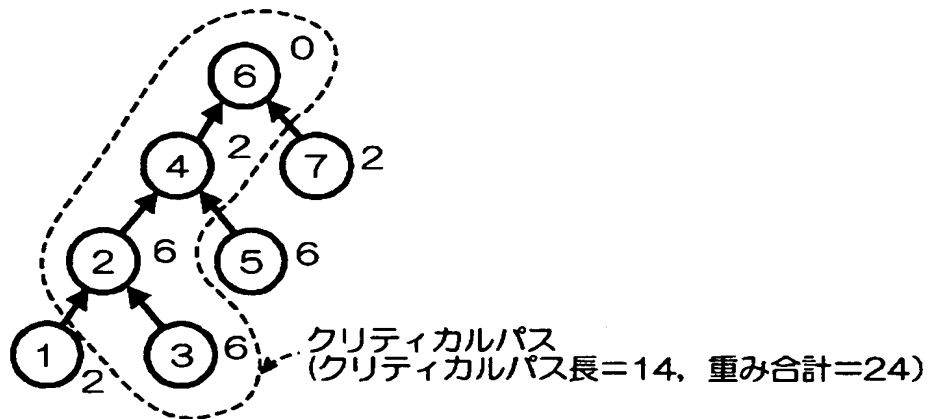
【図11】



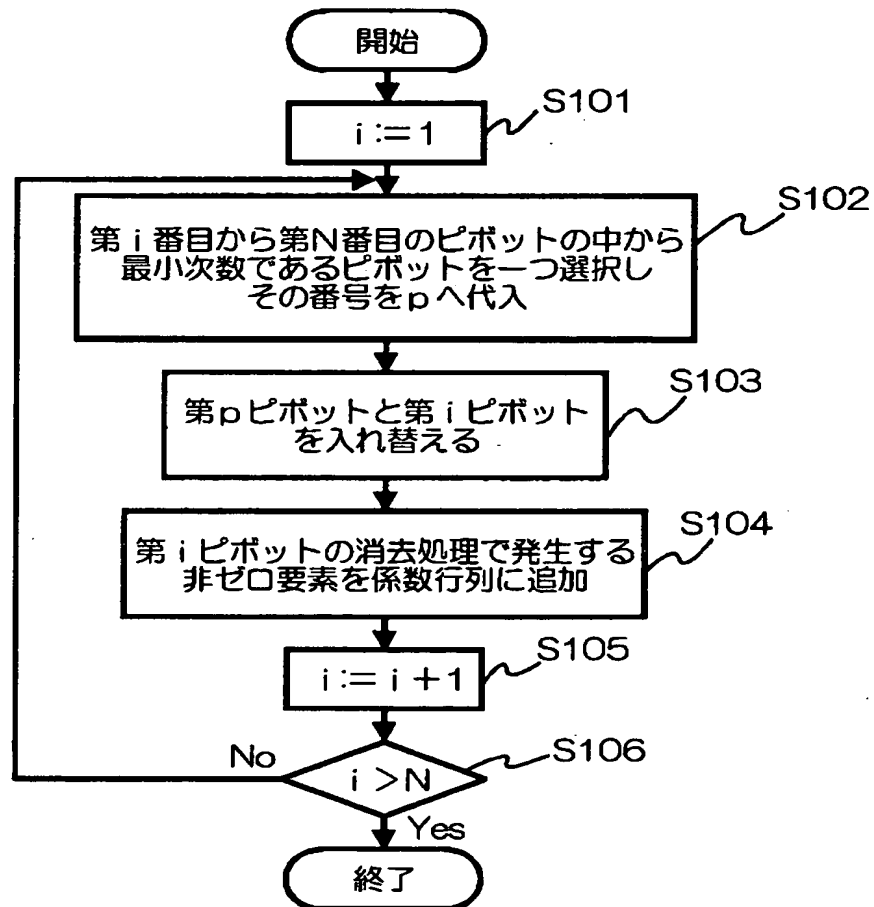
【図12】



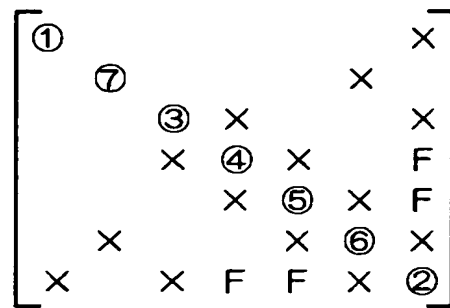
【図13】



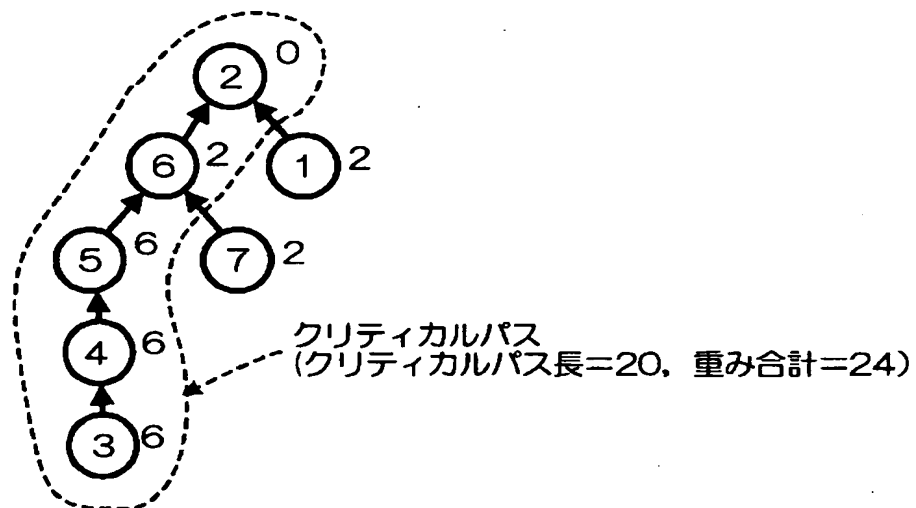
【図14】



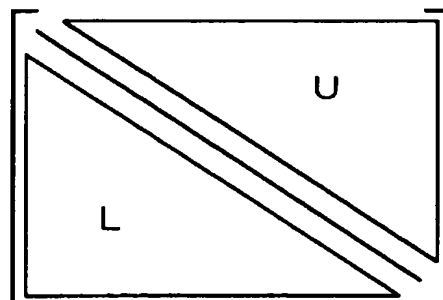
【図 1 5】



【図 1 6】



【図 1 7】



【書類名】 要約書

【要約】

【課題】 ガウス消去処理を複数のプロセッサを用いて並列実行する場合に、従来に比べガウス消去処理の高速化を図る。

【解決手段】 ガウス消去法を電子計算機によって並列実行して解を求める線形連立方程式の係数に対応する構造を有する係数行列の行列順序を決定する行列リオーダーリングを行う際に、 N 行 $\times N$ 列の係数行列中の第 i 番目（ i は $1 \sim N$ ）から第 N 番目までのピボットにおける非ゼロの要素の数に対応する次数 \deg が、最小値 $\min \deg$ に所定の値 α （ α は $1 \sim 3$ ）を加えたもの以下であって、かつ、第 i 番目から第 N 番目までのピボットのうちでクリティカルパス長が最小の第 p 番目のピボットを求め（S 3 0 6）、第 p ピボットと第 i ピボットを入れ替える処理を行う（S 3 0 8）。これを繰り返してすべてのピボットの入れ替えを行うことで行列のリオーダーリングを終了する（S 3 0 4～S 3 1 6）。

【選択図】 図 2

認定・付加情報

特許出願の番号	特願 2 0 0 0 - 1 0 2 1 6 3
受付番号	5 0 0 0 0 4 2 3 9 2 8
書類名	特許願
担当官	高田 良彦 2 3 1 9
作成日	平成 1 2 年 4 月 1 2 日

<認定情報・付加情報>

【特許出願人】

【識別番号】	000004237
【住所又は居所】	東京都港区芝五丁目 7 番 1 号
【氏名又は名称】	日本電気株式会社

【代理人】

【識別番号】	申請人
【識別番号】	100108578
【住所又は居所】	東京都新宿区高田馬場 3 丁目 2 3 番 3 号 O R ビ ル 志賀国際特許事務所
【氏名又は名称】	高橋 詔男

【代理人】

【識別番号】	100064908
【住所又は居所】	東京都新宿区高田馬場 3 丁目 2 3 番 3 号 O R ビ ル 志賀国際特許事務所
【氏名又は名称】	志賀 正武

【選任した代理人】

【識別番号】	100101465
【住所又は居所】	東京都新宿区高田馬場 3 丁目 2 3 番 3 号 O R ビ ル 志賀国際特許事務所
【氏名又は名称】	青山 正和

【選任した代理人】

【識別番号】	100108453
【住所又は居所】	東京都新宿区高田馬場 3 丁目 2 3 番 3 号 O R ビ ル 志賀国際特許事務所
【氏名又は名称】	村山 靖彦

出 願 人 履 歴 情 報

識別番号 [000004237]

1. 変更年月日	1990年 8月29日
[変更理由]	新規登録
住 所	東京都港区芝五丁目7番1号
氏 名	日本電気株式会社